Detecting Malware Infection on Infrastructure Hosted in IaaS Cloud using Cloud Visibility and Forensics

Lama Almadhoor[1]
Department of Computer Science
Jouf University, Sakaka, KSA

A. A. bd El-Aziz[2]
Department of Information Systems
Jouf University, Sakaka, KSA
Department of IST, Cairo University
FGSSR, Egypt

Hedi Hamdi[3]
Department of Computer Science
Jouf University
Sakaka, KSA
University of Manouba, Tunisia

***Abstract*—Cloud computing has been adopted very rapidly by organizations with different businesses and sizes, the use of cloud services is rising at an unparalleled rate these days especially IaaS services as cloud providers offer more powerful resources with flexible offerings and models. This rapid adoption opens new surface attacks to the organizations that attackers abuse with their malware to take advantage of these powerful resources and the valuable data that exist on them. Therefore for organizations to well defend against malware attacks they need to have full visibility not only on their data centers but also on their resources hosted on the cloud and don't take their security for granted. This paper discusses and aims to provide the best approaches to achieve continuous monitoring of malware attacks on the cloud along with their phases (before, during, and after) and the limitations of today's available techniques suggesting needed developments. Logging and forensics techniques have always been the cornerstone of achieving continuous monitoring and detection of malware attacks on-premises, this paper defines the best methods to bring loggings and forensics to the cloud and integrate them with on-premises visibility, thus achieving the full monitoring over the whole security posture of the organization assets whether they are on-premises or on the cloud.**

***Keywords*—Malware attacks; infrastructure as a service (IaaS); amazon web services (AWS); malware detection; cloud forensics; visibility**

I. Introduction

The cloud is a technology that's not new anymore. Nowadays, using cloud services is increasing at an unprecedented pace [1], it has become more popular after the advent of the Fourth Industrial Revolution (IR 4.0) [2] In 2020, about 83% of business workloads operate in the cloud, and a whopping 94% of companies now use a cloud service in one form or shape [3]. There are three most utilized cloud services include Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Infrastructure as a Service (IaaS) is one of the most critical and fastest-growing services in Cloud Computing. As shown in Fig. 1, according to BMC the growth rate from 2018 to 2022 of IaaS is projected to be 18% higher than that of other cloud services.

In this service model, cloud providers provide resources to users/machines such as virtual machines (VMs), raw (block) storage, firewalls, load balancers, and network devices. Resource management for IaaS gives the following advantages Quality of service, scalability, reduced overheads, optimum utility, increased throughput, specialized setting, reduced latency, a streamlined interface, and cost-effectiveness. Virtualization technology is used to offer Infrastructure which enables multiple consumers or tenants to share the same hardware. Virtual machines (VMs) play an important role in Cloud Computing because they allow powerful and systematic use of the hardware available [4].

All these outstanding features make it simple and convenient to access the IaaS service. However, many individuals use this technology in an effective way and few challenges to use it [5]. Infrastructure hosted on the IaaS cloud is becoming targets to many attacks like malware for the following reasons:

1) Cloud service providers steadily offer higher performance with high computation power for their customers. These VMs are big targets for crypto currency mining malware, which are becoming more sophisticated to take the resources of the server without getting noticed.

2) The increase of remote working and globally dispersed workforce and application accessibility especially after the COVID 19 give the attackers more chances to hide their malicious traffic to compromise the cloud-hosted VMs, and use them for their malicious campaigns (phishing campaigns, botnet command, and control, so on).

3) The increase in IoT applications that use cloud-hosted infrastructure to analyze the enormous amounts of data generated by these applications to create business value and insights. Most of these IoT appliances are built with no or weak information security measures thus attackers can easily get their way to the backend cloud-hosted VMs through these IoT devices and applications.

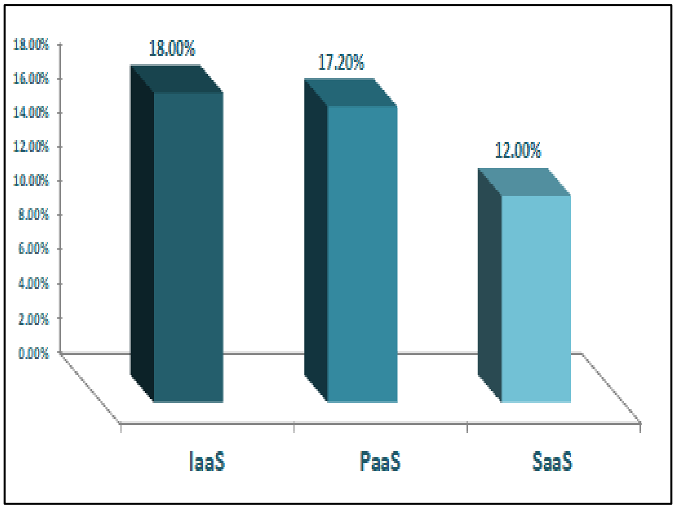


Fig. 1. The Growth Rate (from 2018 to 2022) of cloud Services.

Malware protection is becoming increasingly essential for cloud network security, as it presents a significant threat to network security. Since malware security is minimal, only PC-based protection is up to date [6]. According to Noëlle and others [7] present attacks in the IaaS cloud can be investigated using VMI-based mechanisms. [8] Gives a summary of the different malware detection methods used for intrusion detection. Various machine learning methods can be used to provide a Cloud Computing detection mechanism. In that way, an improved technique is needed to ensure effective intrusion detection for such techniques. The researchers have used a malware detection technique to detect malware in [9], which is SAIDS, a security monitoring system tailored for IaaS clouds, but it only handles one form of IDS. G. Murali and N. Moses [6] in this work propose a framework that deals with anomaly detection malware at the network level, stating that malware distribution in terms of networks varies. Many studies in [10] address cloud forensics in general terms without offering details.

Therefore, this research combines technologies and forensics to monitor and detect malware attacks that targeting VMs in the IaaS cloud. Begin to classify malware attacks in the infrastructure hosted on the IaaS cloud, shed the light on the importance of cloud services visibility to thoroughly track each malicious activity on the cloud-hosted infrastructure and quickly react to malware attack, then list the different approaches and techniques used to gain this visibility and how to perform analytics on this collected data to get insights helping improve the reactive and proactive defenses. Then testing the existing technologies and methodologies for monitoring cloud-hosted infrastructure and performing digital forensics on the cloud-hosted assets and how both can be used to speed up the detection of malware attacks then quickly deploy countermeasures to better stop them from reaching their goals.

This research which addressing the security of infrastructure in IaaS cloud is noble, because it focuses on an investigation that will render the safety of this service, which is conducted on the IaaS cloud. This cloud service has dominated not only small businesses but also global enterprises including big multinationals. As such this topic is very informative since it will detect malware in its early stages. [11] All this provides organizations a feeling of reassurance that their assets are safe and secure.

II. Background

A. Infrastructure as a Service (IaaS)Cloud

The cloud service model's bottom tier is IaaS [12]. It is the most fundamental and critical service, offering basic computing services such as servers, networking, and storage. These resources make use of virtualization technologies to execute services. IaaS also provides users with data security, backup, and maintenance [13, 14]. Consumers have full control over these tools, which are aggregated and controlled [15]. Some companies cannot afford to purchase a computer, so instead of buying the infrastructure, it can be leased or rented according to the needs of the users. This service enhances system availability while also lowering costs and offering a more flexible system.

Common examples of this service: Amazon Web Services (AWS), Google Compute Engine (GCE), Microsoft Azure, and Cisco Metapod.

Despite all of the technical developments in IT security over the last three decades, the Latest statistics also show an increase in malware activity, Deep Instinct conducted analysis and have published a study on the hundreds of millions of attempted cyber-attacks that occurred every day in 2020, Revealing that malware increased by 358 percent overall and Ransom ware increased by 435 percent compared to 2019.

B. Malware Attacks

Malware is a term that combines the word malicious and malware, thus malware is described as software that has a malicious and harmful effect on networks, software, operating systems, or other components [16].

Malware, according to [17], is a software program that is intended to help malicious attackers accomplish their goals. It was created to help attackers accomplish their objectives. Disturbing device processes, altering or hijacking core computing functions and network resources, tracking users' behavior, and stealing, encrypting, or deleting confidential data without the user's permission are just a few of these objectives.

One of the biggest challenges in the IaaS cloud world is malware attacks; malware has long been a major concern to home and business devices, as well as cloud virtual machines [18]. Virtualization, as one of the most important Cloud Computing techniques, blurs the lines between time and space. Virtualization would undoubtedly increase resource efficiency and reduce system management costs [19]. But unfortunately, virtualization creates new vulnerabilities, which are being exploited by malware.

According to Malwarebytes' 2021 Malware Study, attackers exploited the COVID-19 public health crisis in unthinkable ways previously, not only preying on uncertainty and fear during the early months of the global pandemic, but also enhancing malware, retooling assault tactics, and extorting targets to the tune of $100 million. Malware is multiplying at an unprecedented pace, every day AV-TEST registers more than 350,000 new malware in 2021, and there is an increase in the total of malware compare with the last five years as shown in Fig. 2.

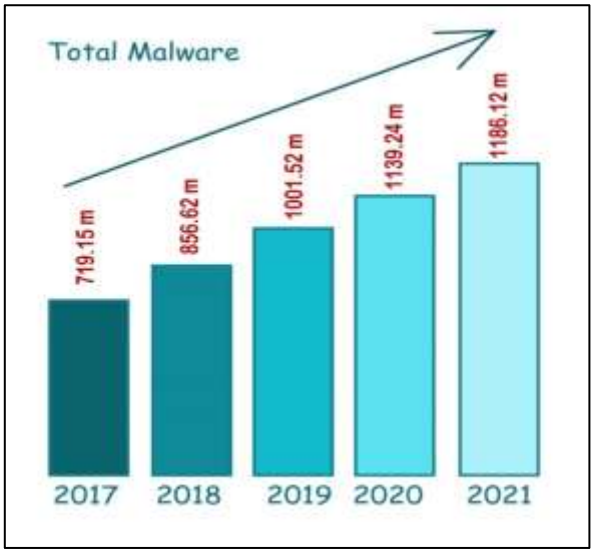


Fig. 2. Total Malware in Last Five Years.

The malware must be discovered before it affects the most resources and networks. In order to protect infrastructure hosted in an IaaS cloud, malware detection must be successful.

C. Malware Detection Methods

In the IaaS cloud world, There are several unique and has different on-demand services available for consumers. These services provide easy and straightforward access to a range of web apps. The virtualized nature of IaaS environments may be a weakness when it comes to malware [20], As a result, malicious attacks and breaches may strike at any time, destroying important apps and files [5]. One of the most difficult challenges in the creation of resilient and stable cloud-based mechanisms is the accurate detection and identification of malware. This is attributed to that malware is often the starting point for phishing, massive Distributed Denial of Service (DDoS) attacks [21], and email spamming by the use of a botnet. For that detecting malware as soon as possible is very important. To resolve the many different attacks and threats that exist in IaaS clouds, continuous and automatic security monitoring of Infrastructure has become a primary requirement [22].

There has been a rapid rise in the number of research studies on malware detection in the IaaS cloud in recent years. A variety of techniques have been proposed to detect malware. These techniques can be divided into four categories: Signature-Based Techniques, Behavior-Based Techniques, cloud-based Techniques, and Machine Learning-Based Techniques.

The procedure for extracting features differs from one Technique to the next. It is difficult to prove that one detection approach is superior to another since each approach has its own set of benefits and drawbacks [23]. Since malware comes in a variety of shapes, sizes, and behaviors, as well as various levels of danger, the same detection methods and mechanisms cannot be used in every situation. It is impossible to detect malware with only one approach to security. [24] As a consequence, having different detection techniques for various conditions is inevitable.

The following describes in detail the approaches to monitor and detect malware attacks:

1) Signature-based techniques for malware detection: A signature is often a short sequence of bytes that is exclusive to each known malware and enables new files to be correctly detected with a low error average [25]. They are commonly used in commercial antivirus software to detect known malware. Traditional signature-based techniques preserve a listing of signatures that are stored in the database, if a match is found, an alert is triggered, and the database is updated [26]. These signatures are formed by disassembling and analyzing the code, there are many disassemblers and debuggers tools are available to aid in the disassembly of portable executable (PE). In this way, code is analyzed then features are extracted. As a consequence, these features play a primary and important role in creating a malware family's signature [27].

Signature-Based techniques are very quick and effective to create a signature. The major feature of this method is the precision with which they detect malware. It is capable of detecting known malware instances quickly and with a smaller amount of malware. A downside of this technique is that of being unable to detect unknown malware. When new malware is released into the market, it must wait until it has infected many systems before its signature can be created and added to the databases. These techniques also have the drawback of being unable to identify encrypted or polymorphic malware [26] that indicates they are not resistant to malware obfuscating techniques.

2) Behavior-based techniques for malware detection: It's also known as Heuristic-Based Detection or anomaly Detection. A behavior-based detection technique uses monitoring tools to observe the programs' behavior and identified whether is it malware or benign? The main goal of this technique is to examine the behavior of malware, known and unknown [23]. It is divided into two phases [28, 29]:

a) The training (learning) phase: Involves monitoring the system's behavior in the absence of an attack or malware, also a learning task is carried out in order to teach the classifier to behave normally.

b) The testing (monitoring) phase: The normal stage is compared with the current behavior then detects suspicious behavior, identifying anomaly activities, looking up the protocols and ports used, and indicate any malicious activities in order to inform the responsible of deviations or major variations from the baseline [30].

The main feature of the behavior-based approach is the capability to detect both unknown malware. However, there are major drawbacks of this technique which are a high False-

Positive Rate (FPR) and a long monitoring duration [31]. Further, the ability to detect malware like zero-day attacks is directly impacted by the decrease of thousands of extracted features, evaluating similarities between them, and surveillance of malware activities [32].

3) Cloud-based techniques for malware detection: Cloud Computing has grown in popularity as a result of its numerous benefits, including easy access, on-demand storage, and lower costs. Because the cloud is so widely used, it's also been used to detect malware. Cloud-based malware detection improves the detection performance for devices and VMs with much larger malware databases and strong and many computational resources. This detection approach employs a variety of detection agents distributed across cloud servers and provides security as a service. The users can upload any type of file and get a record of whether or not the file is malware [23]. In [33] Sun et al created a cloud-based detection malware system, called Cloud Eyes, which has provided resource-constrained devices with effective and trusted security services. Cloud Eyes detected malicious buckets through cross-filtering on the cloud server. In [34] the researchers implemented a malware detection infrastructure realized by an intrusion detection system (IDS) with cloud and mist computing to overcome the IDS sending problem in brilliant objects due to their constrained resources and heterogeneous sub networks.

IaaS cloud is the most flexible model. Users have more choices when it comes to performing IDS over this infrastructure. Intrusion Detection Systems (IDS) can be used in a variety of ways over the IaaS cloud layer [22], including:

4) Network-based Intrusion Detection Systems (NIDS): All network packets are collected and analyzed in the cloud environment using signature or behavior-based detection approaches to identify malicious events and activities like port scanning, DoS attacks, user to root attacks, etc [35-37]. It's used to keep track of network traffic between VMs and host machines, as well as between VMs [22].

There are several models for adapting NIDS to the cloud shown in Table I:

TABLE I. VARIATION BETWEEN CLOUD NIDS/NIPS MODELS

Cloud NIDS model	Ease of management	Cost of operation	Cost of implementation	Customization and Integration capabilities	Level of visibility
IDS on-premises and usage of VPC endpoints	Easy to be manageable as the on-premises IDS scope will just be extended to the cloud environment	High cost due to data transfer expenses as all data will be sent from the cloud environment to the on-premises IDS for inspection. plus, the cost of operating high-performance VPN tunnel	low cost as it doesn’t involve purchasing new appliances or special cloud deployments or modifying network architectures	The IDS rules and policies will be consistent over both the on-premises environment and the cloud. The cloud will just look like an extension to the on-premises environment	The IDS will be able to monitor the traffic going in and out the cloud environment but there is no visibility on inter traffic within the cloud environment itself
IDS on VPC NAT instances or dual-homed systems	Easy to deploy model but hard to be manageable because the organization is responsible for the configuration and customization for each aspect of the IDS and keeping it in pace with new threats and attacks	High cost of operation One single point of failure that can lead to services unavailability when it’s malfunctioned or misconfigured Needs high skilled engineers which can be expensive	The cost of implementing a VPC NAT instance will depend on the size of the VPC and its instances which can be Medium to high cost	Highly Customizable Security teams can add up new features at no cost Highly customizable dashboards and deferent detection mechanisms Due to the high customization, there won’t be obstacles bringing the rules and policies from on-premises IDS, thus providing consistency along both environments	Has visibility on the cloud traffic that will go only through the cloud gateway Doesn’t extend the capabilities of the on-premises IDS
Usage of 3rd party AMI as the NIDS	Custom route and traffic control is required Requires distinct network zone for the appliance for centralized monitoring Management of the appliance is the responsibility of the security team for the organization which adds extra tasks and extra appliances to be managed	Cost will depend the amount of data fed to the appliance As it’s another appliance to be managed, there will be a cost of management will be added in terms of providing the adequate numbers to manage organizations’ appliances	Cost will depend on the Vendor, the features purchased and the amount of data fed to the appliance which will range from Medium to High Cost Due to the high cost If the appliance technology differs from the IDS used on premises, then there will be additional costs for team training or hiring new skills	Customization is limited to the capabilities and features available of the purchased appliance If the technology of the cloud-based IDS differs from the on-premises IDS then there might be rules and polices inconsistency between both environments	Has visibility on the cloud traffic that will go only through the appliance which is to or from the VNets Due to the pay per the data ingested approach organizations usually don’t integrate it with the test and pre-production environments on the cloud

5) Host-based Intrusion Detection Systems (HIDS): At the host level, data is processed, tracked, and analyzed. It monitors and detects modifications in the host kernel, program behavior, and file system [38-40]. These IDS may be installed on a host, virtual machine (VM), or hypervisor (VMM) to detect intrusion events by analyzing device logs against user credentials and access control (AC) policies [41]. In this way, Customers then notify managers if they notice any abnormal activity [22].

The cloud user is in charge of monitoring HIDS deployed on a VM, while the cloud provider is in charge of HIDS deployment on the cloud [42].

6) Distributed Intrusion Detection Systems (IDS): Multiple intrusion detection systems (IDSs) (such as NIDS and HIDS) are deployed over a wide network to track and analyze traffic patterns for intrusion detection, and they can function independently or collaboratively [43,44].

7) VMM/Hypervisor-based Intrusion Detection Systems (HypIDS): Hypervisors that host VMs can easily access performance data; these data offer insight into the activities taking place inside a virtual machine without requiring direct knowledge of the virtual machine's operating system, software, or private data [45].

This type of IDS is installed at the hypervisor layer and monitors and analyzes information transmitted in communications between VMs (i.e., VM-VM), between the VM and the hypervisor, and between the cloud environment and the outside world [46, 47].

8) Machine Learning-Based Techniques Malware Detection: Since Machine learning (ML) can be generalized to never-before-seen malware families and polymorphic strains, machine learning is a common approach to signatureless malware detection [48]. There are a lot of Well-known ML algorithms that particularly useful in behavior-based detection and other detection methods like Artificial Neural Network (ANN), Decision Tree(DT), XGBoost, naive Bayes (NB), Associative Classifier (AC), C4.5 decision tree variant (J48), random forest tree (RF), k-nearest neighbor (KNN), support vector machine (SVM), logistic model trees (LMT), Shared nearest neighbor (SNN), multilayer perceptron (MLP), Bayesian network (BN), simple logistic regression (SLR), RIPPER, Deep Learning (DL), and sequential minimal optimization (SMO) [49-54].

D. Cloud Forensics

Cloud Digital Forensic techniques are typically used to gathering and preserving evidence, reconstructing incidents, deciding how, where, and where an incident happening, and producing threat information. Threat information includes Indicators of compromise that can be used to help an organization defend itself.

III. Methodology

The methodology has been divided into two practical parts:

The First: when the malware attack happened, make cloud analysis for malware detection.

The Second: is Forensics Analysis in the Iaas Cloud after the malware attack happens.

A. Cloud Analysis to Malware Detection

In this practical part, flow many steps as shown in Fig. 3:

1) Choosing test environment: The tests were performed on Amazon Web services (AWS) hosted infrastructure. choosing the Amazon Web services (AWS) for this research because it the market leader for public cloud services offering and has a wide service catalog making it a suitable choice for most organizations, Named as a Leader in Gartner’s Infrastructure as a Service (IaaS) Magic Quadrant for the 7th Consecutive Year. (AWS) innovated many tools and techniques for data collection, monitoring, analysis for their customers which most of the other cloud service providers follow.

2) Data set: Fortunately, there are community initiatives that define and classify each cloud attack technique publicly witnessed; such as the NIST Cybersecurity Framework [55] and MITRE ATT&CK cloud framework.

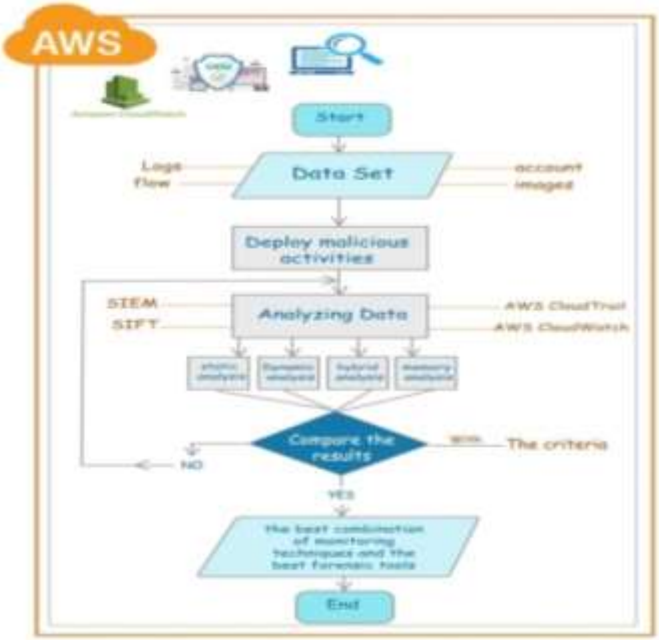


Fig. 3. Flow Chart for Cloud Analysis to Malware Detection.

To define the data to be collected by using the MITRE ATT&CK framework, the combined techniques used to compromise cloud-hosted infrastructure which can be leveraged by the attackers to gain initial access and control over the environment- along with the malware techniques used in enterprise infrastructure. With that combination, the organizations implement continuous monitoring for their infrastructures whether it's on-premises infrastructure or cloud-hosted one. For the scope of research, Continuous monitoring on IaaS can be accomplished by gathering and processing the following [56]:

- API calls Monitoring (In AWS it can be achieved through CloudTrail's logs).
- Host logs and logs of deployed Host Intrusion detection System (HIDs).
- VPC flows.
- Logs of the cloud resources (in AWS it's the CloudWatch Logs) [57].
- Image and instance integrity validation.
- Automation through tools like AWS Lambda and AWS Config.

3) Testing and analysis: Performed the malicious activities performed by the malware without using real malware in this environment.

Use many tools like:

- Amazon CloudWatch: Collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes.
- AWS CloudTrail: A web service that logs your account's AWS API calls and provides you log files.
- AWS Config: provides a comprehensive view of the AWS resource configuration in your AWS account. This involves how the services are connected to one another and how they were previously configured, allowing you to track how the settings and relationships change over time.
- SIEM software: Providing data analysis, event correlation, aggregation, reporting, and log management.
- SIFT is open-source: Includes most tools required for digital forensics analysis and incident response examinations.

Implementation of many testing labs:

a) Create a billing alarm for AWS account: According to the MITRE ATT&CK framework for cloud attacks, one of the most used attack vectors for Cloud attacks and malware attacks targeting cloud-hosted environments is cloud account takeover. There are many ways to detect cloud account takeover, one of the best ways is detecting changes in the usual billing as most cloud malware attacks aim to abuse the environment's resources or deploy new resources. Most public cloud providers provide features to enable their customers to create billing and send them emails when these alarms are triggered

To create a billing alarm for the AWS account that can be used later in detecting any suspicious abuse of IaaS resources, first enable receive billing alerts for the account and then use AWS CloudWatch, to create a metric that will trigger an alarm whenever the billing exceeds a specific threshold.

Also, use the AWS Simple Notification Service (SNS) for sending an email once the alarm is triggered.

Results

When malware misuses the resources of the cloud or publishes expensive new resources, AWS calculates the charges and the estimates charges as per your approach and now have a threshold of 5$ that whenever the charges are exceeded the alarm will be triggered as shown in Fig. 4 and receive an email as a notification.

b) Perform continuous monitoring in the AWS environment: AWS offers a service called AWS Config, this service allows monitoring AWS resource configurations and track resource inventory and changes, which can be used to detect any malicious configuration changes the attacker tries to make to gain control or persistence over the compromised account's resources. This monitoring feeds then can be consumed using AWS CloudWatch and SNS Notifications can be created based on them.

Malware attacks target and modify the data stored and any misconfigured cloud storage leading to leaked data. By using AWS Config to make many rules like sure storage versioning is enabled for AWS storage (S3). By enabling the s3-bucket-versioning-enabled rule, another action performed by attackers is to try to hide their malicious API calls by disabling API calls monitoring, configured a rule to detect if cloudTrail enabled or not and another rule to detect whether the volumes used are encrypted or not. Also to prevent the misuse of the root account, enable another rule to detect whether multi-factor authentication (MFA) is enabled for the root account or not.

In Fig. 5 the Dashboard of AWS Config detected the security issues in resources, which whether indicate a misconfiguration or malicious change.

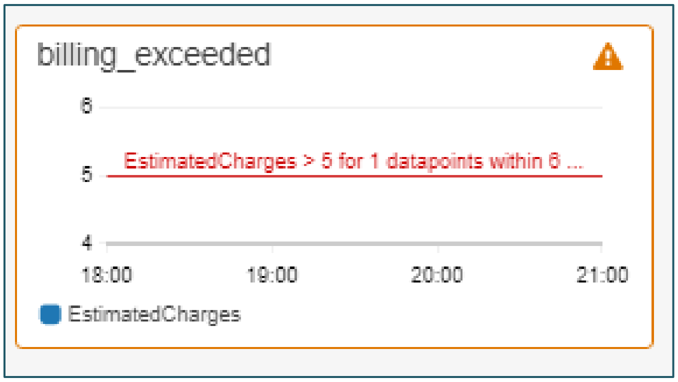


Fig. 4. Alarm for Exceeding Bill for AWS Account.

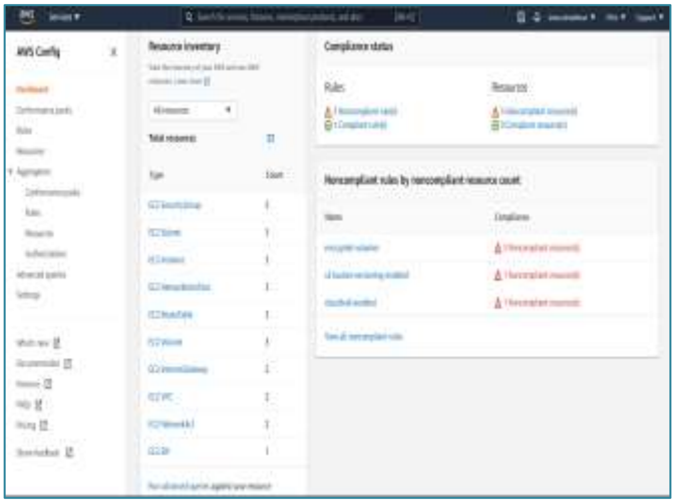


Fig. 5. AWS Config Dashboard.

Using the AWS Config resource inventory to view existing or deleted resources recorded. For a specific resource, view the resource details, configuration timeline, or compliance timeline. The resource configuration timeline allows you to view all the configuration items captured over time for a specific resource. The resource compliance timeline allows you to view compliance status changes. For example, used AWS config to track the timeline of changes for an s3 bucket resource.

Results

- ✓ By using AWS Config detected Malware attacks that modify the data stored, and target any misconfigured cloud storage.
- ✓ By using rules in AWS Config can see if malware attacks try to hide their malicious API calls by disabling API calls monitoring.

Limitations

- ✗ Not all monitoring options are available for each region, cloud providers may offer their services to a specific region but monitoring these services may not be available for this region or came late.
- ✗ The AWS config service takes a few minutes to rebuild asset inventory, thus not providing real-time monitoring.

c) Monitoring cloud API Calls: In AWS CloudTrail is used to monitor account activity and API calls, this is a very important feature as cloud providers offer their services via APIs. CloudTrail feeds can be integrated with CloudWatch to create metrics generating alarms for any suspicious account's behavior or any account misuse.

Create a trail; I enable ongoing delivery of events as log files to an Amazon S3 bucket. Then get the logs and API Calls from CloudTrail Event history.

Fig. 6 detects console logins (recording the account used and the source IP that the user has when he logged in) which reveals any attempts to login from suspected places or countries.

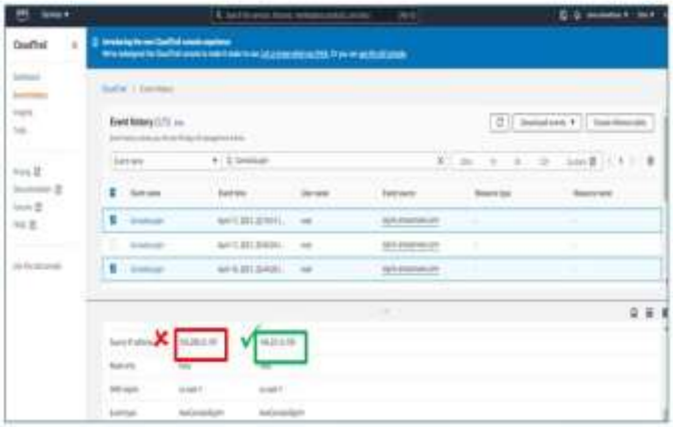


Fig. 6. Detecting Console Logins in AWS CloudTrail.

Also can find that AWS generated logs are formatted in JSON.

CloudTrail has a lot of great, very detailed log data. also, By setting up a CloudTrail trail I deliver CloudTrail events to Amazon S3 then review this in the console or view/download from S3.

Results

- ✓ Detect console logins from suspected places or countries.
- ✓ Because most of the time, organizations transfer cloud logs to their own data center for long term storage and correlation with other on-premises tools, it's very important to generate the logs in a text-like format such as JSON, thus enabling serialization of complex, high-quality log data and decouple the interpretation of logs from specific solution or vendor. From a test, notice AWS uses this concept in their generated logs and flows.
- ✓ Leverage the storage API (s3 API) to import cloud trails to a search and indexing platform or security management systems like (SIEM solution) for building more unified and robust use cases monitoring the security posture over the entire environment.

Limitations

- ✗ By default, there are no trails configured to log any of AW'S activities. Must enable what wants to monitor.
- ✗ CloudTrail Logs directly in the console is not as efficient as pulling the logs into some tool for parsing and normalization with better searching and filtering features.

d) Generating VPC network flows (VPC Flows): Producing flow logs of communications that occur in the VPC (north-west communications and east-west communications) is very crucial to detect activities related to malware attacks such as communications with malicious IPs and command and control servers, attempts of pivoting and lateral spreading, and suspicious communications behaviors and data exfiltration.

CloudWatch, created a flow log group to watch for flows within VPC [58] while one instance is performing lateral scanning, to detect the scan.

By generating these flows and sending those to CloudWatch then create metrics watching for specific suspicious behaviors or policy violation communications. For example, made a metric to watch for Telnet communications that can due to policy violation or scanning attempt and received an alarm as shown in Fig. 7.

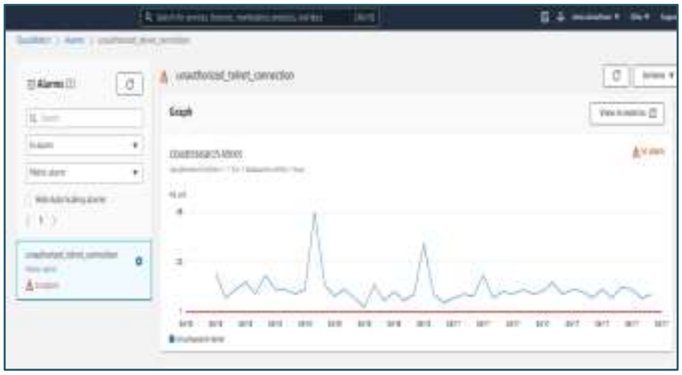


Fig. 7. Alarm for unauthorized Telnet Communication.

Also, receive this notification by email.

Results

- ✓ Seeing all flow logs of communications that occur in the VPC.
- ✓ By sending flow logs to CloudWatch then create metrics watching for specific suspicious behaviors or policy violation communications.
- ✓ Instead of sending flow logs to CloudWatch, can send them to s3 bucket storage to be aggregated with other log sources and offloaded to on-premises with Storage API calls and leveraging them using SIEM Solution.

Limitations

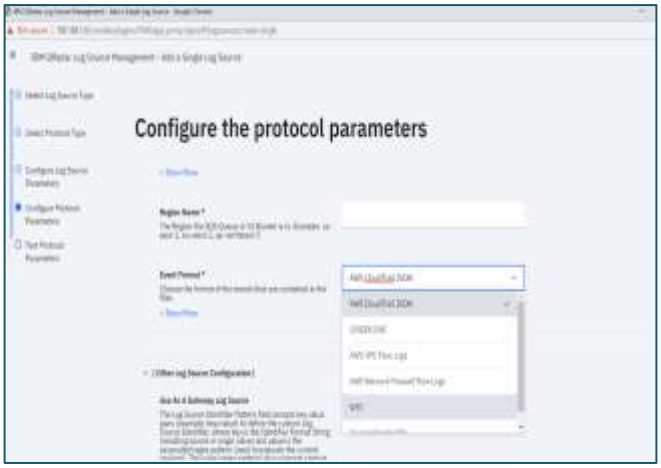
- × Flow logs may take up to 5 minutes to appear, thus not providing real-time visibility.
- × Reviewing VPC Flow Logs directly in the console is not efficient while using CloudWatch Metrics to make use of these flows, leveraging the high potential of captured flows is achieved through sending them to some security event management tool like SIEM Solution for more flexible correlation and enrichment.

e) Transferring logs to SIEM: Must bringing cloud logs into a single point where they can be aggregated with on-premises security events and other security and intelligence feeds, thus enabling the threat management team to have a single pane of glass from which they can monitor the whole security posture of their organization.

To achieve this purpose, install IBM Qradar Community edition (SIEM Solution) locally and configure it to receive the AWS CloudTrail logs which configured earlier to monitor different services of the AWS account and its resources [56].

1) The first thing to notice that there are many protocol options to bring the CloudTrails Logs to SIEM solution.

2) Fig. 8 contains the different types of event formats that can be stored in s3 bucket (the important ones are AWS CloudTrail and AWS VPC Flow logs).



Configure the protocol parameters

Fig. 8. Setting Options to Select Event Formats that Stored in S3 Bucket.

3) Specify the S3 bucket name and the directory where AWS CloudTrails resides.

4) The maximum frequency to poll the logs from the AWS s3 bucket is 1 minute, providing an adequate timeframe to detect attacks but still not real-time visibility.

Results

The different ways to transfer Flow logs from the cloud to threat management like SIEM is shown in Table II.

- ✓ For most IaaS use cases, the best way to transfer monitoring logs to on-premises threat management tools like SIEM is through the usage of Storage APIs, yet there are other methods suitable for the less common of today's use cases.
- ✓ Also, a statistic in Fig. 9 determines the number of logs that are generated per second from resources on the cloud, it helps us to determine the amount of visibility that gets from the logs, for example, NGFW and HDIS have many activities, so they give me the greatest amount of visibility and helps us see all the events in real-time and discover malware quickly.

B. Forensics Analysis in The Iaas Cloud

Using cloud forensics to assist companies in enhancing their incident response and threat detection capabilities [59] organizations must have sufficient forensics investigation expertise to apply to their cloud infrastructure to ascertain the root cause of an attack, detect signs of vulnerability, and better protect against IaaS malware attacks, as well as quickly locate malware and its objectives before they have an impact on the companies' operations.

In the event of a hacked virtual machine, several cloud users automatically terminate and destroy the virtual machine (VM), erasing all proof in the process.

TABLE II. THE BEST WAYS TO TRANSFER FLOW LOGS FROM THE CLOUD TO THREAT MANAGEMENT

	SysLog or SysLog Server	**Storage API**	**Data Stream**
Data transfer Speed	It's not considered a real-time data communication. Data is ready to be sent as soon It's generated or collected by the aggregation server. Because in most cases it's not compressed, it will take few minutes to arrive at the premises,	Faster than sending Syslog messages, but still will take time for the data to be written in the storage bucket. Fast but not real-time data communication.	Data processed as soon it's generated. Because it makes use of data analytics features the data transferred is small and fast. it provides real-time visibility. Example AWS Kinesis streams
Cost	High cost around 9 cents per Gigabyte	-Low cost around .007 cent per Gigabyte. -There will be additional cost for the storage bucket	Expensive service
Size of logs transferred	If there are big number of resources and no compression used, the size of logs will be big leading to more costs	High quality compressed data generated small size data transfers	Rich and Small size of data streams
Data enrichment and preprocessing	No analysis or processing is made before sending	Better quality data and fully inflated data to flows and logs	performs data analytics, correlation and enrichment with other feeds, sending only good quality actionable insights
Scalability	Not scalable as the volume of data increases	Scalable as it uses API communication methods	Scalable
Log Data protection	By default, data is sent in plain text unless sent over VPN connection or syslog collector used	the connection is encrypted	Communication is encrypted
Need for additional tools or management	For optimum experience VPN tunnel and syslog collector are used	configure additional storage to store the logs	Purchasing the data analytics stream service

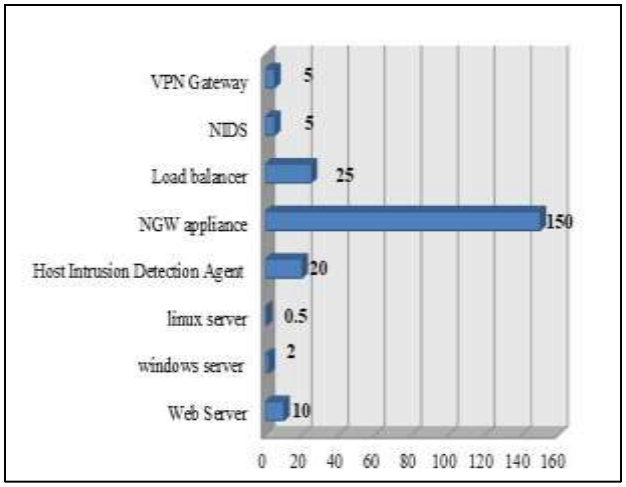


Fig. 9. Logs Generated from Cloud Resources Per Second.

It can be difficult to plan for forensics in the cloud. Until recently, there have been few tools to assist analysts in inspecting applications and collecting data [60]**.** When it comes to gathering and analyzing evidence, must look for the following:

- Network packet captures (PCAPs) for network forensics.
- Memory for instance.
- A disk for instance.
- Event data and logs.

Recently, more vendors and community members have been concentrating their efforts on resolving forensics issues in the cloud. How do we best gather evidence? How do we store it properly? What tools work in the cloud well?

1) Evidence Capture

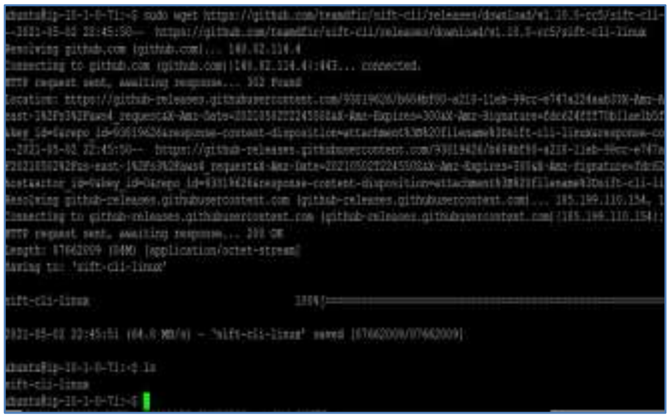
- Capture a disk is getting easier in a running instance. You take a snapshot of EBS (Elastic Block Store) in Amazon EC2 (Amazon Elastic Compute Cloud) [61], after that, attach it to a forensic workstation (covered in a moment). The Azure, You can grab IaaS OS and Data drives directly from the portal.
- Capturing memory in a shared environment would necessitate some form of per-instance basis capture. To put it another way, instances' running memory would have to be acquired using separate tools (local or remote).

2) Test environment preparation

3) Provisioning the Forensic machine and installing the required forensics investigation tools. Fortunately, a package that provides access to most of the forensics tools from one executable package is called SIFT. Prepare forensics investigation machine as follows:

a) create an instance that will be used to perform the investigation. In a test, named it cloudresearch-forensics-instance.

b) SSH into the cloudresearch-forensics-instance to download the SIFT tools as shown in Fig. 10 [in research time the latest SIFT version was 1.10.0-rc5].



```
ubuntu@ip-10-1-0-71:~$ sudo wget https://github.com/teamdfir/sift-cli/releases/download/v1.10.0-rc5/sift-cli-
--2021-05-02 22:45:50--  https://github.com/teamdfir/sift-cli/releases/download/v1.10.0-rc5/sift-cli-linux
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/[illegible]
[illegible]
[illegible]
--2021-05-02 22:45:50--  https://github-releases.githubusercontent.com/[illegible]
[illegible]
[illegible]
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.110.154, 1
Connecting to github-releases.githubusercontent.com (github-releases.githubusercontent.com)|185.199.110.154|:
HTTP request sent, awaiting response... 200 OK
Length: 67662009 (64M) [application/octet-stream]
Saving to: 'sift-cli-linux'

sift-cli-linux                       100%[==================================================

2021-05-02 22:45:51 (64.0 MB/s) - 'sift-cli-linux' saved [67662009/67662009]

ubuntu@ip-10-1-0-71:~$ ls
sift-cli-linux
ubuntu@ip-10-1-0-71:~$
```

Fig. 10. Download the SIFT Tools.

c) For ease of access move the executable file to the account binaries directory

```
mv ./sift-cli-linux /usr/local/bin/sift

chmod +x /usr/local/bin/sift
sudo sift install
```

1) Create a snapshot from the instance to perform forensics analysis on it.

2) Create a volume from it with the snapshot in the same available zone as cloudresearch-forensics-instance.

3) The Evidence Volume will be available to be attached to cloudresearch-forensics-instance to perform investigation.

4) Then attaché the evidence cloudresearch-forensics-instance by follow:

From volume actions we selected Attach Volume and choose to attach it to cloudresearch-forensics-instance as shown in Fig. 11 [62].

We confirmed the successful attachment from the console as shown in Fig. 12.

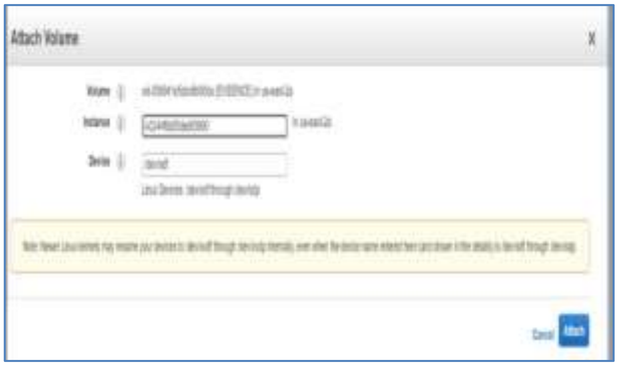


Fig. 11. Attaching the Evidence to the SIFT Workstation.

Fig. 12. Successful Attachment.

We verified that using lsblk command from cloudresearch-forensics-instance

```
$ sudo  lsblk
NAME               MAJ:MIN      RM      SIZE       RO      TYPE
MOUNTPOINT
Xvda             202:0          0       8G       0      disk
  └─Xvda1        202:1          0       8G       0      part        /
Xvdf             202:80         0       8G       0      disk
  └─Xvdf1        202:81         0       8G       0      part
```

The *xvdf* disk is attached volume with a single partition not mounted yet.

We determined the format of partition using ***file*** command

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo file –s /dev/xvdf1

/dev/xvdf1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

As we are analyzing a Linux machine, we can see that the format is XFS.

Mount the evidence Linux file system.

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo mount -o ro /dev/xvdf1 /mnt/evidence /
```

Verify a successful mount of the evidence volume using ***ls*** command.

```
$sudo  ls  –als /mnt/evidence/
total 20
dr-xr-xr-x    18   root     root      257    Apr    14   17:37     .
drwxr-xr-x   18   root     root     4096   May     4    05:45     ..
-rw-r--r--      1   root     root     0         Apr   14   17:37    .autorelabel
lrwxrwxrwx  1   root     root     7         Mar   26    17:35    bin –>
usr/bin
dr-xr-xr-x     4   root     root     4096   Mar   26     17:36    boot
drwxr-xr-x    3   root     root     136      Mar   26     17:36    dev
drwxr-xr-x   81  root     root     8192   Apr   17     04:05    etc
drwxr-xr-x    3   root     root     22        Apr    14     17:37     home
lrwxrwxrwx  1   root     root     7          Mar    26    17:35    lib ->
usr/lib
lrwxrwxrwx   1   root     root       9        Mar    26    17:35    lib64 ->
usr/lib64
drwxr-xr-x      2   root     root       6        Mar    26    17:35    local
drwxr-xr-x      2   root     root       6        Apr    9       2019    media
drwxr-xr-x      2   root     root       6        Apr    9       2019    mnt
drwxr-xr-x      4   root     root      27        Mar    26     17:36    opt
drwxr-xr-x      2   root     root       6         Mar    26     17:35    proc
dr-xr-x---       3   root     root     103        Apr   14     17:37    root
drwxr-xr-x      3   root     root      18         Mar   26     17:36    run
```

```
lrwxrwxrwx   1   root     root     8        Mar   26    17:35    sbin ->
usr/sbin
drwxr-xr-x     2   root     root      6        Apr    9      2019    srv
drwxr-xr-x     2   root     root      6        Mar   26    17:35    sys
drwxrwxrwx   8   root     root      172     May   4      03:54    tmp
drwxr-xr-x     13 root     root      155      Mar   26    17:35    usr
drwxr-xr-x     19 root     root       269       Apr  14    17:37    var
Ubuntu@cloudresearch-forensics-instance:~
```

5) Forensic analysis on linux EC2 instance: Fig. 13 contains perform common disk image forensic investigation domain exercises on the snapshot of EC2 instance's volume and define any limitations and special configurations needed to facilitate the investigation [63].

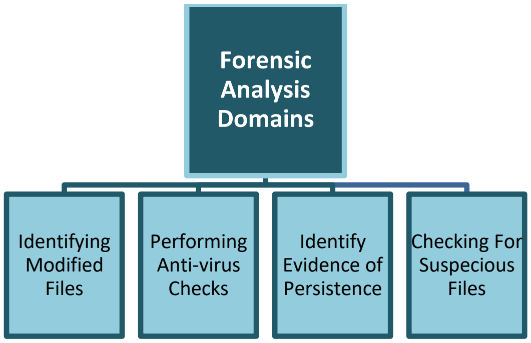

Forensic Analysis Domains
Identifying Modified Files
Performing Anti-virus Checks
Identify Evidence of Persistence
Checking For Suspecious Files

Fig. 13. Forensic Analysis Domains Test.

d) Identifying modified and added files in the invested file system:

1) Create a new instance from the same type of instance to perform forensics analysis on. This new instance will be used as a reference to a known good state. Achieve that by taking a snapshot of the newly created instance, mounting it to the cloudresearch-forensics-instance in read-only then creating hashes of all files on the reference volume to help identify the differences between the Evidence and the Baseline.

```
Ubuntu@cloudresearch-forensics-instance:~

$ lsblk

NAME          MAJ:MIN   RM   SIZE    RO  TYPE     MOUNTPOINT
Loop0            7:0         0    55.5M   1    loop       /snap/core18/1988
 Loop1           7:1         0    33.3M   1    loop       /snap/amazon-ssm-
Loop2            7:2         0    55.5M   1      loop      /snap/core18/1997
Loop3            7:3         0    70.4M   1      loop       /snap/lxd/19647
Loop4            7:4         0    32.3M   1      loop       /snap/snapd/11588
Loop5            7:5         0    31.1M   1      loop        /snap/snapd/11036
Loop6            7:6         0    69.9M   1      loop        /snap/lxd/19188
Xvda             202:0      0     8G       0      disk
  └─ Xvda1      202:1      0     8G       0      part       /
xvdf             202:80     0     8G       0      disk
  └─ xvdf1      202:81     0     8G       0      part       /mnt/evidence
xvdg             202:96     0     8G       0      disk
  └─ xvdg1      202:97     0     8G       0      part       /mnt/baseline
```

2) creating a hash database of all Baseline Volume files as follows:

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo find /mnt/baseline/ -type f –exec /usr/bin/md5sum {}\; >
baseline_files.md5
```

3) Then making a hash of all files on the volume under investigation. And compare them with the reference file hashes, storing them in changed_files.txt

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo find /mnt/evidence/ -type f –exec /usr/bin/md5sum {}\; >
investigate_files.md5
```

```
Ubuntu@cloudresearch-forensics-instance:~

$ ls –al *.md5

-rw-rw-r--   1    ubuntu  ubuntu   4304670    May   6   15:18
basline_files.md5

-rw-rw-r--   1    ubuntu  ubuntu   3865228    May   6   15:15
investigate_files.md5
```

4) After making a hash index of the both files using ***hfind*** command, the 2 hash tables to find the difference. then make the output in a text file.

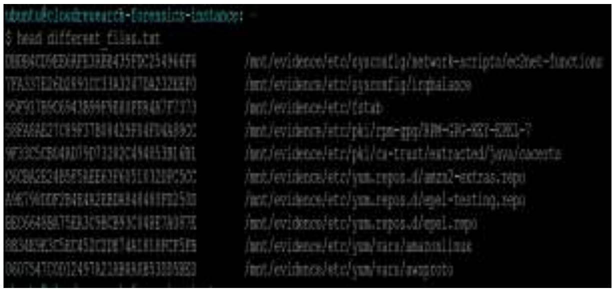
```
Ubuntu@cloudresearch-forensics-instance:~

$ awk '{print $1}' investigate_files.md5 | hfind baseline_files.md5 |
grep "Hash Not Found" | awk '{print $1}' > difference.md5
```

```
Ubuntu@cloudresearch-forensics-instance:~

$ hfind –f difference.md5 investigate_files.md5 > different_files.txt
```

Fig. 14 shows a sample of Files that has been changed.



```
Ubuntu@cloudresearch-forensics-instance:~
$ head different_files.txt
[illegible]        /mnt/evidence/etc/sysconfig/network-scripts/ec2net-functions
[illegible]        /mnt/evidence/etc/sysconfig/irqbalance
[illegible]        /mnt/evidence/etc/fstab
[illegible]        /mnt/evidence/etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
[illegible]        /mnt/evidence/etc/pki/ca-trust/extracted/java/cacerts
[illegible]        /mnt/evidence/etc/yum.repos.d/amzn2-extras.repo
[illegible]        /mnt/evidence/etc/yum.repos.d/epel-testing.repo
[illegible]        /mnt/evidence/etc/yum.repos.d/epel.repo
[illegible]        /mnt/evidence/etc/yum/vars/amazonlinux
[illegible]        /mnt/evidence/etc/yum/vars/awsproto
```

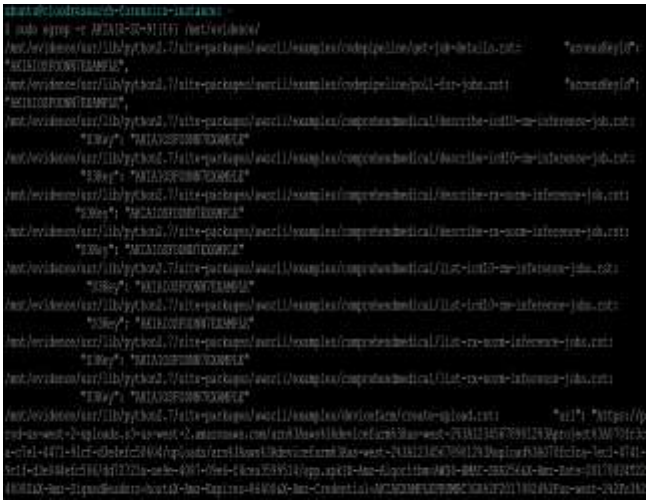
Fig. 14. Sample of Output.

5) Finding keys on the compromised system: Private keys are usually found in hidden directories and the SSH by analysts. Looking for private keys in SSH and AWS hidden directories (.ssh/ and .aws/). In an EC2 case, unprotected private keys are a bad security practice that could be violating the company's security policy. If any private SSH keys or AWS keys are present, they must be supposed to be compromised.

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo ls /mnt/evidence/home/ec2-user/.ssh/authorized_keys
```

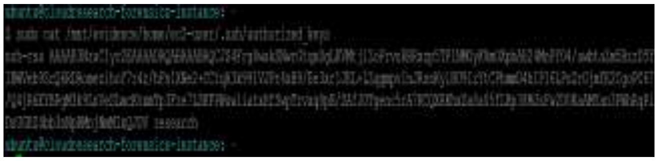
Sample of the command searching for the pattern AKIA[A-Z0-9] shown in Fig. 15.



```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo egrep -r "AKIA[A-Z0-9]{16}" /mnt/evidence/
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/codepipeline/get-job-details.rst:       "accessKeyId":
"AKIAIOSFODNN7EXAMPLE",
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/codepipeline/poll-for-jobs.rst:       "accessKeyId":
"AKIAIOSFODNN7EXAMPLE",
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/describe-icd10-cm-inference-job.rst:
            "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/describe-icd10-cm-inference-job.rst:
            "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/describe-rx-norm-inference-job.rst:
           "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/describe-rx-norm-inference-job.rst:
           "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/list-icd10-cm-inference-jobs.rst:
             "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/list-icd10-cm-inference-jobs.rst:
             "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/list-rx-norm-inference-jobs.rst:
            "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/comprehendmedical/list-rx-norm-inference-jobs.rst:
            "KMSKey": "AKIAIOSFODNN7EXAMPLE"
/mnt/evidence/usr/lib/python2.7/site-packages/awscli/examples/devicefarm/create-upload.rst:          "url": "https://p
rod-us-west-2-uploads.s3-us-west-2.amazonaws.com/[illegible]
[illegible]
[illegible]
[illegible]
```

Fig. 15. Sample of Output from Command AKIA [A-Z0-9] Pattern.

Another common practice is to look at the list of public keys used to SSH into the instance (home/*/.ssh/authorized keys), which can be used in conjunction with syslog messages to figure out who has accessed the system as shown in Fig. 16.



```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo cat /mnt/evidence/home/ec2-user/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC[illegible]
[illegible]
[illegible]
[illegible] research
ubuntu@cloudresearch-forensics-instance: ~
```

Fig. 16. List of Public Keys those are used to SSH to the Instance.

e) Performing Anti-Virus checks

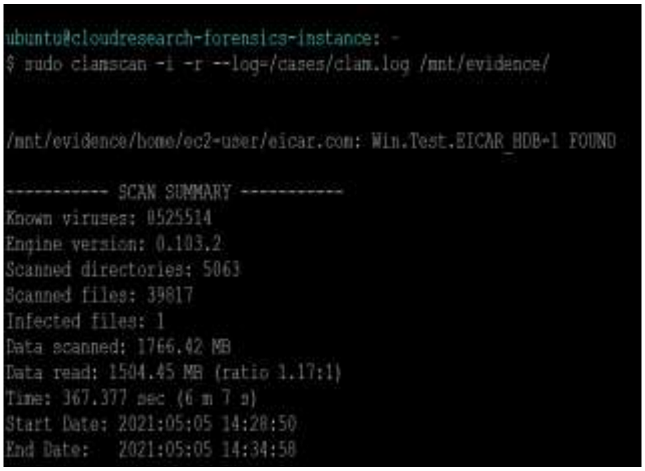
1) Find any security software installed: Most cloud providers provide a mechanism for central management of VMs' systems deployed on the IaaS. These systems are used to patch, configure and audit the VMs to a baseline. Then information available through these systems may support the forensics investigation. For AWS this system is called AWS System Manager. If the AWS system Manager is present its default location will be /usr/bin/amazon-ssm-agent and its log path will be /var/log/amazon/ssm/amazon-ssm-agent.log. Another tool that can be offered by the Cloud provider to the deployed VMs is vulnerability scanner. AWS offers AWS Inspector. If this tool was deployed on the compromised machine indicates that there may be vulnerability data available via the console which can help focus the investigation.

2) Scan the image with antivirus: To validate if malware files can be detected by using AntiVirus Scan tools, download EICAR file to an investigated instance before taking a snapshot of its volume. EICAR is ANTI MALWARE Test file. The scanning of a snapshot volume has no different than scanning conventional forensics images and able to detect the EICAR file. For this purpose, use ClamAV that comes with the SIFT package. Then scan the mounted evidence with ClamAV.

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo clamscan –i –r --log=/cases/clam.log /mnt/evidence/
```

The previous command results are shown in Fig. 17.



```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo clamscan -i -r --log=/cases/clam.log /mnt/evidence/


/mnt/evidence/home/ec2-user/eicar.com: Win.Test.EICAR_HDB-1 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 8525514
Engine version: 0.103.2
Scanned directories: 5063
Scanned files: 39817
Infected files: 1
Data scanned: 1766.42 MB
Data read: 1504.45 MB (ratio 1.17:1)
Time: 367.377 sec (6 m 7 s)
Start Date: 2021:05:05 14:28:50
End Date:   2021:05:05 14:34:58
```

Fig. 17. Result of Scan the Image with Antivirus.

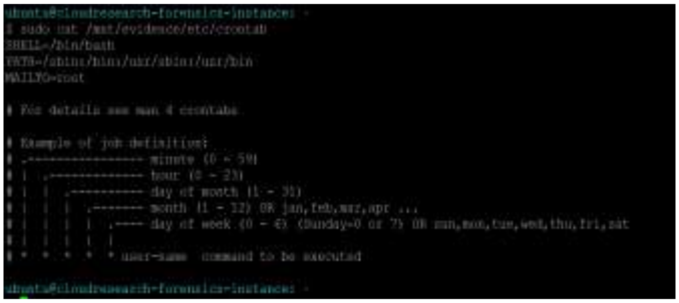
f) Identify Evidence of Persistence

The malware must use a persistence mechanism to survive a reboot. The two most popular methods are start-up scripts and cron jobs.

1) View the cron task by viewing crontab file and listing cron files as shown in Fig. 18:

```
Ubuntu@cloudresearch-forensics-instance:~

cat /mnt/evidence/etc/crontab
```



```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo cat /mnt/evidence/etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

ubuntu@cloudresearch-forensics-instance: ~
```

Fig. 18. Snapshot of Corntab File and Corn Files.

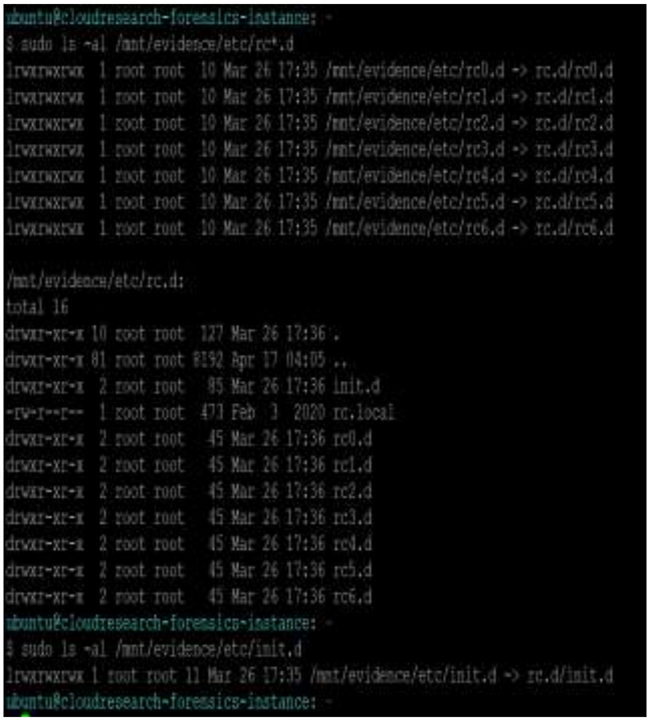
Also check the corn jobs for all users.

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo ls /mnt/evidence/etc/cron.*
```

2) Looking for unusual start-up scripts: While entering a specific run stage, some malwares can use the start-up scripts that Linux runs at boot time. These scripts can be found in

/etc/init.d on some Linux distributions, but they will be in /etc/rc*.d on Amazon Linux as shown in Fig. 19:



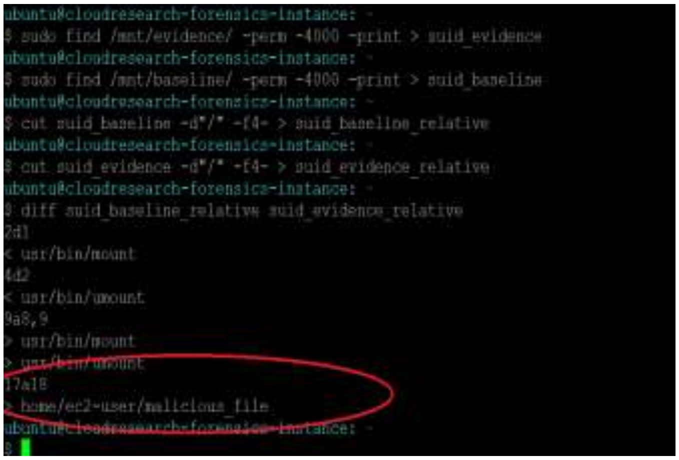
```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo ls -al /mnt/evidence/etc/rc*.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc0.d -> rc.d/rc0.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc1.d -> rc.d/rc1.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc2.d -> rc.d/rc2.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc3.d -> rc.d/rc3.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc4.d -> rc.d/rc4.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc5.d -> rc.d/rc5.d
lrwxrwxrwx  1 root root  10 Mar 26 17:35 /mnt/evidence/etc/rc6.d -> rc.d/rc6.d

/mnt/evidence/etc/rc.d:
total 16
drwxr-xr-x 10 root root  127 Mar 26 17:36 .
drwxr-xr-x 81 root root 8192 Apr 17 04:05 ..
drwxr-xr-x  2 root root   85 Mar 26 17:36 init.d
-rw-r--r--  1 root root  473 Feb  3  2020 rc.local
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc0.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc1.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc2.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc3.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc4.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc5.d
drwxr-xr-x  2 root root   45 Mar 26 17:36 rc6.d
ubuntu@cloudresearch-forensics-instance: ~
$ sudo ls -al /mnt/evidence/etc/init.d
lrwxrwxrwx 1 root root 11 Mar 26 17:35 /mnt/evidence/etc/init.d -> rc.d/init.d
ubuntu@cloudresearch-forensics-instance: ~
```

Fig. 19. Snapshot of Start-up Scripts on Amazon Linux.

g) Checking for Suspicious Files: Perform all techniques related to looking and searching for suspicious files as perform them in the conventional forensic analysis. As follows:

- Look for the contents of the */tmp* directory and unusual *SUID* files. To search for unusual SUID files, make a base line using baseline volume and compared it with the volume under investigation. Purposefully add a malicious _file that has suid permission enabled to the home directory of the ec2-user account on the compromised machine and successfully to detect it as shown in Fig. 20:



```
ubuntu@cloudresearch-forensics-instance: ~
$ sudo find /mnt/evidence/ -perm -4000 -print > suid_evidence
ubuntu@cloudresearch-forensics-instance: ~
$ sudo find /mnt/baseline/ -perm -4000 -print > suid_baseline
ubuntu@cloudresearch-forensics-instance: ~
$ cut suid_baseline -d"/" -f4- > suid_baseline_relative
ubuntu@cloudresearch-forensics-instance: ~
$ cut suid_evidence -d"/" -f4- > suid_evidence_relative
ubuntu@cloudresearch-forensics-instance: ~
$ diff suid_baseline_relative suid_evidence_relative
2d1
< usr/bin/mount
4d2
< usr/bin/umount
9a8,9
> usr/bin/mount
> usr/bin/umount
17a18
> home/ec2-user/malicious_file
ubuntu@cloudresearch-forensics-instance: ~
$
```

Fig. 20. Detect Malicious _file in Home Directory of the ec2-user Account.

- Use the same method to look for large files above a certain size limit and compare them to the base volume.
- Use the ***strings*** command to get a fast indication of the file's existence and to spot possible signs of compromise. The strings command's output can include IP addresses, file names, and configuration information that expose the malware's intent.

```
Ubuntu@cloudresearch-forensics-instance:~

$ string /mnt/evidence/home/ec2-user/eicar.com

X50!P%@AP[4\PZX45(P^)7CC)7}$EICAR-STANDARD-
ANTIVIRUS-TEST-FILE!$H+H*
```

- Use 3rd party tools like **ViruTotal** to investigate the suspected files detected by this exercise and validate if the samples have been seen in the wild as shown in Fig. 21.

For this purpose, calculate the SHA256 hash of detected malware (EICAR) and then look for it on VirusTotal.

```
Ubuntu@cloudresearch-forensics-instance:~

$ sudo sha256sum /mnt/evidence/home/ec2-user/eicar.com

275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf6
51fd0f /mnt/evidence/home/ec2-user/eicar.com
```

C. Discussion and Area of Improvements

A common mistake is to let snapshots be shared publicly (across different AWS accounts), without great care is taken, this can lead to leakage of sensitive data and keys.

According to Amazon, the only way to capture the memory of an Amazon Linux EC2 instance is via an SSH session, passing the SSH keys that have the root privilege to a remote memory imaging tool.

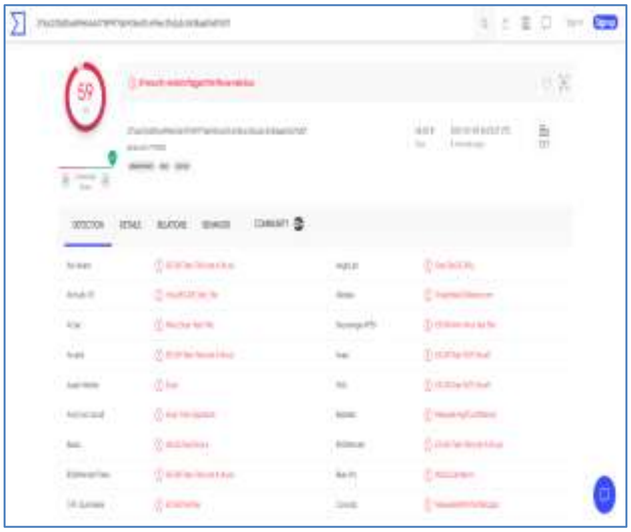


Fig. 21. Virustotal Result for the Malware we Detected on Machine.

As a best practice incident handler should provide a new forensics analysis machine for each case. Provisioning the forensics analysis machine on AWS takes some time, unlike the conventional on-prem machines. But one way to speed this up is to make an AMI (Amazon Machine Image) of the SIFT.

The volumes created from snapshots must be in the same availability zone of the forensics analysis workstation.

There is a quiet difference between conventional on-premises evidence gathering and evidence gathering from IaaS virtual machines and due to the need for rapid containment of the compromised cloud virtual machine, these evidence gathering processes must be integrated with infrastructure automated deployments tools for automating the evidence gathering.

Memory analysis for IaaS virtual machines is very challenging as most leading solutions for image analysis require a profile that is specific to the kernel of the system that was imaged, so they have profiles for operating systems used on-premises but don't have any profiles for the Cloud provided Linux images. Custom profiles can only be made for these tools if only the analyst has the source code for the kernel headers of these cloud-provided Linux images.

Amazon Linux differs from traditional on-prem Linux especially in the path for some services and files such as the path for the startup scripts and the path of all cron jobs, thus creating an inconsistency between different cloud providers Linux distributions and the forensic analyst have to learn the customizations before beginning his investigation.

The audit logs generated by the Amazon Linux process which is by default enabled will take some practice to read and learn how to read them.

For instances running web servers, the logs of the web service exist in unusual places on the volume, thus the best way to analyze these logs is to pull them into a log server or SIEM solution.

Using cloud web load balancers such as Amazon's Elastic Load Balancer may hide the address of the attacker if no special customization to the load balancer configuration.

EC2 instances are set to UTC by default, thus if anyone trying to make a timeline for malware attacks that spread from the cloud-hosted infrastructure to the on-premises infrastructure, to make a TimeZone correction before correlating cloud instances logs with on-premises logs.

Cloud instances use a default setting for Syslog that sometimes is inconsistent with the common or Syslog settings used on-premises. For example, Amazon Linux images in the marketplace use a default setting for Syslog that does not include the year in the date stamp. Ideally, a cloud administrator uses a configuration template for configuring new instances on the cloud which properly configures the Syslog timestamp to include the year, but this may not be the case with each organization or cloud administrator.

The challenge for forensics in the cloud is usually "Will it meet the chain of custody requirements?" and "Will it hold up in court?" You need to enable write-once storage that is owned solely by the forensics and IR teams and carefully document your IAM policy for the IR/Forensics role. In addition, ALL activity around evidence acquisition and evidence storage location should be logged extensively.

Despite standardized, well-defined, and matured tools of Incident Response and evidence acquisition for suspected on-premises assets, there are no standardized processes for Digital forensics automation on the cloud and very few tools have been built to do this for the cloud today. Almost nothing is available commercially and each cloud service provider tries to provide its own solution for this purpose for example Microsoft has centered most of its security capabilities around Security Center and Sentinel within Azure.

IV. Conclusion

Recently, the number, severity, sophistication of malware attacks, and cost of malware infect on the world economy have been increasing exponentially. Malware should be detected before damaging the important assets in the company. In this research, we applied cybersecurity and security in-depth principles to the cloud IaaS environment. These principles embrace that defense controls will fail at some point and an attack will succeed so organizations must have response mechanisms to put off these attacks as soon as possible, to be able to detect and monitor the attack while and post its occurrence is very crucial. Log monitoring and digital artifacts gathering are the cornerstone enablers for monitoring and detection of active malware attacks. In this research, we defined a practical baseline of the security telemetry that needs to be configured on cloud IaaS environments to maintain continuous visibility and monitoring. We achieved the baseline of the practical part through using of MITRE ATT&CK framework for Cloud which classifies cloud attacks and cloud attack vectors that have been abused so far. After that, we validated the applicability and limitation of deploying this baseline using the AWS environment. Collecting logs without performing investigation and analysis of these logs will provide no help regarding the attack detection. This can be achieved by using security event management tools such as SIEM solutions that perform data correlation, enrichment, integration with other security events, and long-term storage. In the second part of the research, review the different ways to transfer logging data from IaaS environment to the SIEM solution located on-premises. The third part of this research investigated the applicability of performing Digital Forensic Investigations on the compromised IaaS VMs. In future work, we suggest that cloud providers should provide the maintenance tools for performing volatile memory analysis for their VMs. Also, develop a new automated tool for incident response and forensics investigation on the IaaS.

References

[1] B. Varghese and R. Buyya, "Next Generation Cloud Computing: New Trends and Research Directions," *Elsevie :Future Generation Computer Systems*, Vol. 79, pp. 1-22, September 2017.

[2] W. Yassin, M. F. Abdollah, R. Ahmad, Z. Yunos and A. Ariffin, "Cloud Forensic Challenges and Recommendations: A Review," *OIC-CERT Journal of Cyber Security*, vol.2, issue.1, pp. 19 – 29, February 2020.

[3] W. Dawoud , I. Takouna and C. Meinel, "Infrastructure as a Service Security: Challenges and Solutions," *Conference: Informatics and*

Systems (INFOS), 2010 The 7th International Conference, IEEE Explore., April 2010.

[4] R. V. Deshmukh and K. K. Devadkar, "Understanding DDoS Attack & Its Effect In Cloud Environment," *Procedia Computer Science, Elsevier*, Vol.49, pp. 202 – 210, 2015.

[5] A. Bedi, N. Pandey and S. K. Khatri, "Analysis of Detection and Prevention of Malware in Cloud Computing Environment," *2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE*, pp. 918-921, 2019.

[6] N. Babu and G. Murali, "Malware Detection for Multi Cloud Servers using Intermediate Monitoring Server," *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), IEEE*, pp. 3609-3612, Aug 2017.

[7] N. Rakotondravony, B. Taubmann, W. Mandarawi, E. Weishäupl, P. Xu, B. Kolosnjaji, M. Protsenko, H. de Meer and H. P. Reiser, "Classifying malware attacks in IaaS cloud environments," *Journal of Cloud Computing:Advances, Systems and Applications*, Vol.6, no.26, December 2017.

[8] H. Rathore, S. Agarwal, S. K. Sahay and M. Sewak, "Malware Detection using Machine Learning and Deep Learning," *International Conference on Big Data Analytics , Springer,* LNCS, Vol. 11297, pp. 402-411, 4 Apr 2019.

[9] A. Giannakou, L. Rilling, C. Morin and . J.-L. Pazat, "SAIDS: A Self-Adaptable Intrusion Detection System for IaaS Clouds," *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 354-355, May 2018.

[10] A. Alenezi, R. K. Hussein, R. J. Walt and G. B. Wills, "A Framework for Cloud Forensic Readiness in Organizations," *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) ,IEEE Xplore*, 12 June 2017.

[11] I. Ahmad and H. Bakht, "A Novel Detection and Prevention (DAP) Framework for Abuse of cloud service threat," *International Journal of Computing and Network Technology*, Vol.6, Issue.3, no.3, Sept 2018.

[12] S. Fatima and S. Ahmad, "An Exhaustive Review on Security Issues in Cloud Computing," *Ksii Transactions On Internet And Information Systems*, Vol. 13,Issue. 6, pp. 3219-3237, Jun 2019.

[13] K. Lokuge, "Security Concerns in Cloud Computing: A Review," *Secure Software Development Assignment, Researchgate*, December 2020.

[14] B. Mohammed, B. Modu, K. M. Maiyama, H. Ugail and I. Awan, "Failure Analysis Modelling in an Infrastructure as a Service (Iaas) Environment," *Electronic Notes in Theoretical Computer Science*, Vol.340, p. 41–54, October 2018.

[15] E. B. Chawkia, A. Ahmeda and T. Zakariae, "IaaS Cloud Model Security Issues on Behalf Cloud Provider and User Security Behaviors," *Procedia Computer Science,* Vol.134, pp.328–333, January 2018.

[16] Nancy, S. Silakari and U. Chourasia, "A Survey Over the Various Malware Detection Techniques used in Cloud Computing," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 5 Issue.02, pp. 398-402, February 2016.

[17] Y. YE, T. LI, D. ADJEROH and S. S. IYENGAR, "A Survey on Malware Detection Using Data Mining Techniques," *ACM Computing Surveys*, Vol. 50, No. 3, pp. 1-41, June 2017.

[18] X. Gao, C. Hu, C. Shan, B. Liu, Z. Niu and H. Xie, "Malware classification for the cloud via semi-supervised transfer learning," *Journal of Information Security and Applications, Elsevier*, Vol.55, 20 October 2020.

[19] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang and L. Liu, "Fault tolerant scheduling for real-time scienti_c work_ows with elastic resource provisioning in virtualized clouds," *IEEE Transactions on Parallel and Distributed Systems*, Vol.27, Issue.12 pp. 3501 - 3517, , Dec. 2016.

[20] A. K. Marnerides, M. R. Watson, N. Shirazi, A. Mauthe and D. Hutchison, "Malware analysis in Cloud Computing: Network and system characteristics," *IEEE Globecom Workshops (GC Wkshps)*, December 2013.

[21] M. R. Watson, N.-u.-h. Shirazi, A. K. Marnerides, A. Mauthe and D. Hutchison, "Malware Detection in Cloud Computing Infrastructures," *Transactions on Dependable and Secure Computing, IEEE*, Vol.13, Issue.2, pp.192-205, January 2015.

[22] M. Abdelsalam, R. Krishnan and R. Sandhu, "Clustering-Based IaaS Cloud Monitoring," *2017 IEEE 10th International Conference on Cloud Computing (CLOUD*), June 2017.

[23] Ö. ASLAN and R. SAMET, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access,* Vol.8 , pp. 6249 - 6271, 03 January 2020.

[24] N. Praveena, S. Sofia and D. Srinivasulu, "Anomaly Detection in Infrastructure Service of Cloud Computing," *International Journal of Computer Science Trends and Technology (IJCST)*, Vol.4, Issue.6, pp. 104-108, Nov - Dec 2016.

[25] Y. Ye, T. Li, S. Zhu, W. Zhuang, E. Tas, U. Gupta and M. Abdulhayogl, "Combining file content and file relations for cloud based malware detection," *the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 222–230, August 2011.

[26] P. Srivastava and M. Raj, "Feature extraction for enhanced malware detection using genetic algorithm," *International Journal of Engineering & Technology*,Vol.7, pp. 444-449, March 2018.

[27] M. Jain and P. Bajaj, "Techniques in Detection and Analyzing Malware Executables: A Review," *International Journal of Computer Science and Mobile Computing*, Vol. 3, Issue. 5, pp. 930 – 935, May 2014.

[28] R. Sihwail, K. Omar and K. A. Ariffin, "A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis," *International Journal on Advanced Science Engineering and Information Technology*, Vol.8, No.4-2, pp. 1662-1671, September 2018.

[29] A. Damodaran, F. D. Troia, C. A. Visaggio, T. H. Austin and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," *Journal of Computer Virology and Hacking Techniques* , Vol.13, pp.1-12, 29 December 2015.

[30] J. D. Araújo and Z. Abdelouahab, "Virtualization in Intrusion Detection Systems: A Study on Different Approaches for Cloud Computing Environments*," IJCSNS International Journal of Computer Science and Network Security,* Vol.12 No.11, pp. 9-16, November 2012.

[31] Z. Bazrafshan, H. Hashemi, S. M. Hazrati Fard and A. Hamzeh, "A survey on heuristic malware detection techniques," *5th Conference on Information and Knowledge Technology (IKT*), *IEEE,* pp. 113-120, May 2013.

[32] M. H. M. Yusof and M. R. Mokhtar, "A Review of Predictive Analytic Applications of Bayesian Network," *International Journal on Advanced Science Engineering and Information Technology*, Vol.6, No.6, pp. 857-867, August 2016.

[33] H. Sun, X. Wang, R. Buyya and J. Su, "CloudEyes: cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices," *Software Practice and Experience*, Vol. 47, June 2016.

[34] S. Shen, L. Huang, H. Zhou, S. Yu, E. Fan and Q. Cao, "Multistage Signaling Game-Based Optimal Detection Strategies for Suppressing Malware Diffusion in Fog-Cloud-Based IoT Networks," *IEEE Internet of Things Journal*, Vol.5, Issue.2, pp. 1043 - 1054, April 2018.

[35] H. Hamad, M. Hoby, "Managing Intrusion Detection as a Service in Cloud Networks," *International Journal of Computer Applications* , Vol.41, No.1 pp. 35-40, March 2012.

[36] D. Singh, D. Patel, B. Borisaniya and C. Modi, "Collaborative IDS Framework for Cloud," *International Journal of Network Security*, Vol.18, No.4, pp. 699-709, July 2016.

[37] C. N. Modi and D. Patel, "A novel hybrid-network intrusion detection system (H-NIDS) in Cloud Computing," *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, April 2013.

[38] K. Vieira, A. Schulter, C. B. Westphall and C. Merkle , "Intrusion Detection for Grid and Cloud Computing," *IEEE IT Professional* , Vol.12, Issue.4, pp. 38 - 43, July-Aug. 2010.

[39] J.-H. Lee, . M.-W. Park, J.-H. Eom and T.-M. Chung, "Multi-level Intrusion Detection System and log management in Cloud Computing," *13th International Conference on Advanced Communication Technology (ICACT2011), IEEE*, Feb 2011.

[40] J. Arshad, P. Townend and J. xu, "An Abstract Model for Integrated Intrusion Detection and Severity Analysis for Clouds," *International*

Journal of Cloud Applications and Computing, Vol.1, pp. 1-16, January 2011.

[41] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Elsevier journal-Computer Science Review* ,Vol.33, pp. 1–48, 2019.

[42] Y. Mehmood, U. Habiba, M. A. Shibli and R. Masood, "Intrusion Detection System in Cloud Computing: Challenges and opportunities," *IEEE 2nd National Conference on Information Assurance (NCIA)*, December 2013.

[43] C.-C. Lo, C.-C. Huang and J. Ku, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks," *IEEE 39th International Conference on Parallel Processing Workshops*, Sept 2010.

[44] A. V. Dastjerdi, K. Abu Bakar and S. G. H. Tabatabaei, "Distributed Intrusion Detection in Clouds Using Mobile Agents," *Third International Conference on Advanced Engineering Computing and Applications in Sciences*, Oct 2009.

[45] J. Nikolai and Y. Wang, "Hypervisor-based cloud intrusion detection system," *International Conference on Computing, Networking and Communications (ICNC)*, *IEEE*, February 2014.

[46] N. Pandeeswari and G. Kumar, "Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN," *Mobile Networks and Applications*, Vol.21, pp. 494–505, 16 August 2015.

[47] A. Aldribi, I. Traoré, B. Moa and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Elsevier: Computers & Security*, Vol.88, January 2020.

[48] H. S. Anderson, A. Kharkar and B. Filar, "Evading Machine Learning Malware Detection," *Black Hat USA*, pp. 22-27, July 2017.

[49] H. El Merabet and A. Hajraoui, "A Survey of Malware Detection Techniques based on Machine Learning," *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 1, pp. 366-373, 2019.

[50] P. HarshaLatha and R. Mohanasundaram, "Classification Of Malware Detection Using Machine Learning Algorithms: A Survey," *International Journal Of Scientific & Technology Research* , Vol. 9, Issue. 02, pp. 1796-1802, February 2020.

[51] B. Cakir and E. Dogdu, "Malware Classification Using Deep Learning Methods," *ACMSE '18: Proceedings of the ACMSE 2018 Conference*, March 2018 .

[52] S. Fathima, S. and A. K. , "Comparative Analysis of Malware Classification using Machine Learning Algorithms," *International Journal of Engineering Research and Applications*, Vol. 10, Issue 12, pp. 64-68, December 2020.

[53] E. Raff, J. Barker, J. Sylvester and R. Brandon, "Malware Detection by Eating a Whole EXE," *The Workshops of the Thirty-Second AAAI Conference on Artificial Intelligence,* pp. 268-276, October 2017.

[54] L. W. B. S. Y. B. &. Z. Q. X. Liu, "Automatic malware classification and new malware detection using machine learning," *Frontiers of Information Technology & Electronic Engineering*, Vol.18, pp. 1336-1347, 27 October 2017.

[55] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," *NIST Special Publication:800-144*, December 2011.

[56] B. Balakrishnan and R. R. Varuni, "Cloud Security Monitoring," *SANS Institute*, 8 Mar 2017.

[57] A. Amazon Web Services, *Amazon CloudWatch Developer Guide*, 2010.

[58] B. Beach, S. Armentrout , R. Bozo and E. Tsouris, "Virtual Private Cloud," *in Pro PowerShell for Amazon Web Services*, USA, Apress, Berkeley, CA, pp. 85-115, 22 September 2019.

[59] J. Dykstra , "Digital forensics for infrastructure-as-a-service cloud computing," Ph.D dissertation, *Faculty of the Graduate School of the University of Maryland, Baltimore County*, 2013.

[60] A. Pichan, M. Lazarescu and S. T. Soh, "Cloud forensics: Technical challenges, solutions and comparative analysis," *Digital Investigation ,Elsevier,* Vol.13, pp. 38-57, 23 March 2015.

[61] A. Anand, "Managing Infrastructure in Amazon using EC2,CloudWatch, EBS, IAM and CloudFront," *International Journal of Engineering Research & Technology (IJERT),* Vol.6, Issue.03, pp. 373-378, March 2017.

[62] Д. Ранделович, К. Кук, В. Боровик, Д. Младенович and Д. Эрлевайн, "Influence Of The Operating System On The Forensics Tools," *Thematic Conference proceedings Archibald Reiss*, Vol.343, 2016.

[63] B. Carrier, *File system forensic analysis*, US: Pearson Education, Inc., 2005.